ДЗ по "Теория вероятностей и математическая статистика (семинары)"

Семинар 8. Корреляционный анализ

**Задание 1**

Даны значения величины заработной платы заемщиков банка (zp) и значения их поведенческого кредитного скоринга (ks):

zp = [35, 45, 190, 200, 40, 70, 54, 150, 120, 110],

ks = [401, 574, 874, 919, 459, 739, 653, 902, 746, 832].

Найдите ковариацию этих двух величин с помощью элементарных действий, а затем с помощью функции cov из numpy

Полученные значения должны быть равны. Найдите коэффициент корреляции Пирсона с помощью ковариации и среднеквадратичных отклонений двух признаков, а затем с использованием функций из библиотек numpy и pandas.

Ковариаця:

Где  – ковариация величин zp и ks;

M() – математическое ожидание.

коэффициент корреляции Пирсона:

где – ковариация величин zp и ks;

– стандартное отклонение величин zp и ks.

**Код:**

from statistics import mean

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from scipy import stats

from math import sqrt

ROUND\_NUMBER = 2

zp = np.asarray([35., 45, 190, 200, 40, 70, 54, 150, 120, 110])

ks = np.asarray([401., 574, 874, 919, 459, 739, 653, 902, 746, 832])

cov\_zp\_ks = mean(zp\*ks)-mean(zp)\*mean(ks)

print(f'Задание 1.1: Расчетная ковариация: {round(cov\_zp\_ks,ROUND\_NUMBER)}')

print(f'Задание 1.1: numpy:                {np.cov(ks,zp, ddof=0)}')

cof\_cor\_pearson = cov\_zp\_ks/(np.std(zp, ddof=0)\*np.std(ks, ddof=0))

print(f'Задание 1.2: Расчетный коэф-т корреляции Пирсона: {round(cof\_cor\_pearson,ROUND\_NUMBER)}')

print(f'Задание 1.2: numpy:                {np.corrcoef(ks,zp)}')

df = pd.DataFrame({'zp': [35., 45, 190, 200, 40, 70, 54, 150, 120, 110],

 'ks': [401., 574, 874, 919, 459, 739, 653, 902, 746, 832]})

pandas\_cof\_cor\_pearson = df['zp'].corr(df['ks'], method='pearson')

print(f'Задание 1.2: pandas:              {round(pandas\_cof\_cor\_pearson, ROUND\_NUMBER)}')

plt.scatter(df['zp'], df['ks'])

plt.show()

**Результат:**

Задание 1.1: Расчетная ковариация: 9157.84

Задание 1.1: numpy: [[30468.89 9157.84]

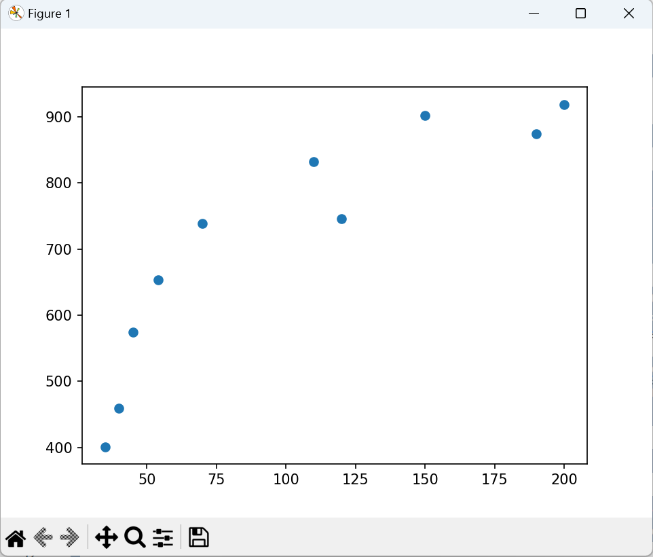
[ 9157.84 3494.64]]

Задание 1.2: Расчетный коэф-т корреляции Пирсона: 0.89

Задание 1.2: numpy: [[1. 0.88749009]

[0.88749009 1. ]]

Задание 1.2: pandas: 0.89



**Задание 2**

Измерены значения IQ выборки студентов, обучающихся в местных технических вузах:

131, 125, 115, 122, 131, 115, 107, 99, 125, 111.

Известно, что в генеральной совокупности IQ распределен нормально.

Найдите доверительный интервал для математического ожидания с надежностью 0.95.

Решение:

Где – среднее выборочное;

Z – z критерий (используем z по сколько известно что у нас генеральная совокупность, в противном случае используем t критерий), находим по таблице или z=stats.norm.ppf(1-.05/2);;

α – интервал, который мы отбрасываем;

Ϭ – среднеквадратичное отклонение ;

n – объем выборки.

**Код:**

iq = [131, 125, 115, 122, 131, 115, 107, 99, 125, 111]

iq\_std = np.std(iq)

print(f'Промежуточные данные iq\_std: {round(iq\_std,ROUND\_NUMBER)}')

iq\_mean = mean(iq)

print(f'Промежуточные данные iq\_mean: {round(iq\_mean,ROUND\_NUMBER)}')

z = stats.norm.ppf(1-.05/2)

print(f'Промежуточные данные z: {round(z,ROUND\_NUMBER)}')

point\_1 = iq\_mean - z\*iq\_std/sqrt(len(iq))

point\_2 = iq\_mean + z\*iq\_std/sqrt(len(iq))

print(f'Задание 2: Доверительный интервал:[{round(point\_1, ROUND\_NUMBER)}; {round(point\_2, ROUND\_NUMBER)}]')

**Результат:**

Задание 2: Доверительный интервал: [111.9; 124.3]

**Задание 3**

Известно, что рост футболистов в сборной распределен нормально с дисперсией генеральной совокупности, равной 25 кв.см. Объем выборки равен 27, среднее выборочное составляет 174.2. Найдите доверительный интервал для математического ожидания с надежностью 0.95.

Решение:

Где – среднее выборочное;

Z – z критерий (используем z по сколько известна дисперсия генеральной совокупности, в противном случае используем t критерий), находим по таблице ;

α – интервал, который мы отбрасываем;

Ϭ – среднеквадратичное отклонение

Где D – дисперсия генеральной совокупности;

n – объем выборки.

**Код:**

lon\_mean = 174.2

lon\_std = sqrt(25)

lon\_n = 27

z = stats.norm.ppf(1-.05/2)

# print(f'Промежуточные данные z: {round(z,ROUND\_NUMBER)}')

dot\_1 = lon\_mean - z\*lon\_std/sqrt(lon\_n)

dot\_2 = lon\_mean + z\*lon\_std/sqrt(lon\_n)

print(f'Задание 3: Доверительный интервал:[{round(dot\_1, ROUND\_NUMBER)}; {round(dot\_2, ROUND\_NUMBER)}]')

**Результат:**

Задание 3: Доверительный интервал: [172.31; 176.09]